Predicting Stroke - PreProcessing

Stephen Kuc

2022-06-20

```
stroke <- read.csv("c:/Users/steph/OneDrive/Documents/USD/ADS503/healthcare-dataset-stroke-data.csv")
```

```
library(caret) # for training models
```

```
## Warning: package 'caret' was built under R version 4.1.3
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

```
## Loading required package: lattice
```

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.1.3
```

```
library(Hmisc)
```

```
## Warning: package 'Hmisc' was built under R version 4.1.3
```

```
## Loading required package: survival
```

```
##
## Attaching package: 'survival'
```

```
## The following object is masked from 'package:caret':
##
##     cluster
```

```
## Loading required package: Formula
```

```
## Warning: package 'Formula' was built under R version 4.1.1
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following object is masked from 'package:e1071':
##
##     impute

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.1.3

## corrplot 0.92 loaded
```

```
library(plyr)
```

```
## Warning: package 'plyr' was built under R version 4.1.3

##
## Attaching package: 'plyr'

## The following objects are masked from 'package:Hmisc':
##
##     is.discrete, summarize
```

```
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 4.1.3

## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
# changing datatypes to what they should be
stroke$hypertension <- as.factor(stroke$hypertension)
stroke$heart_disease <- as.factor(stroke$heart_disease)
stroke$gender <- as.factor(stroke$gender)
stroke$ever_married <- as.factor(stroke$ever_married)
stroke$work_type <- as.factor(stroke$work_type)
stroke$Residence_type <- as.factor(stroke$Residence_type)
stroke$smoking_status <- as.factor(stroke$smoking_status)
stroke$bmi <- as.numeric(stroke$bmi)
```

```
## Warning: NAs introduced by coercion
```

```
stroke$stroke <- as.factor(stroke$stroke)
```

Getting rid of ID column, centering, scaling and imputing

```
stroke <- subset(stroke,select = -c(id))
stroke_imp <- preProcess(stroke,method = c("center", "scale", "knnImpute"))
strokePp <- predict(stroke_imp, stroke)
```

Converting multi level factors into dummy variables to check correlations and save for models that need dummy variables

```
# convert multi level factors into dummy variables
strokeFact <- subset(strokePp, select = c(gender, work_type, Residence_type, smoking_status))
strokeNon <- subset(strokePp,select = -c(gender,work_type, Residence_type, smoking_status))

dmy <- dummyVars(" ~. ", data =strokeFact)
strokeDum <- data.frame(predict(dmy,newdata=strokeFact))
head(strokeDum)
```

```
##   gender.Female gender.Male gender.Other work_type.children work_type.Govt_job
## 1             0           1            0                   0                   0
## 2             1           0            0                   0                   0
## 3             0           1            0                   0                   0
## 4             1           0            0                   0                   0
## 5             1           0            0                   0                   0
## 6             0           1            0                   0                   0
##   work_type.Never_worked work_type.Private work_type.Self.employed
## 1                      0                1                       0
## 2                      0                0                       1
## 3                      0                1                       0
## 4                      0                1                       0
## 5                      0                0                       1
## 6                      0                1                       0
##   Residence_type.Rural Residence_type.Urban smoking_status.formerly.smoked
## 1                    0                    1                              1
## 2                    1                    0                              0
## 3                    1                    0                              0
## 4                    0                    1                              0
## 5                    1                    0                              0
## 6                    0                    1                              1
##   smoking_status.never.smoked smoking_status.smokes smoking_status.Unknown
## 1                           0                     0                      0
## 2                           1                     0                      0
## 3                           1                     0                      0
## 4                           0                     1                      0
## 5                           1                     0                      0
## 6                           0                     0                      0
```

```
# add back into dataframe with multi-level factors now as dummy variables

strokeDummies <- cbind(strokeDum, strokeNon)
head(strokeDummies)
```

```
##   gender.Female gender.Male gender.Other work_type.children work_type.Govt_job
## 1             0           1            0                  0                  0
## 2             1           0            0                  0                  0
## 3             0           1            0                  0                  0
## 4             1           0            0                  0                  0
## 5             1           0            0                  0                  0
## 6             0           1            0                  0                  0
##   work_type.Never_worked work_type.Private work_type.Self.employed
## 1                      0                1                        0
## 2                      0                0                        1
## 3                      0                1                        0
## 4                      0                1                        0
## 5                      0                0                        1
## 6                      0                1                        0
##   Residence_type.Rural Residence_type.Urban smoking_status.formerly.smoked
## 1                    0                    1                              1
## 2                    1                    0                              0
## 3                    1                    0                              0
## 4                    0                    1                              0
## 5                    1                    0                              0
## 6                    0                    1                              1
##   smoking_status.never.smoked smoking_status.smokes smoking_status.Unknown
## 1                           0                    0                      0
## 2                           1                    0                      0
## 3                           1                    0                      0
## 4                           0                    1                      0
## 5                           1                    0                      0
## 6                           0                    0                      0
##         age hypertension heart_disease ever_married avg_glucose_level
## 1 1.0513314            0             1          Yes       2.706110617
## 2 0.7859932            0             0          Yes       2.121350940
## 3 1.6262309            0             1          Yes      -0.005027809
## 4 0.2553167            0             0          Yes       1.437217451
## 5 1.5820079            1             0          Yes       1.501037522
## 6 1.6704540            0             0          Yes       1.768021830
##           bmi stroke
## 1  0.98124492      1
## 2  0.28606366      1
## 3  0.45922236      1
## 4  0.70113526      1
## 5 -0.62301952      1
## 6  0.01359335      1
```

Going to change the remaining binary factor variables (hypertension, heart_disease, and ever married) to numerical 1, 0 variables.

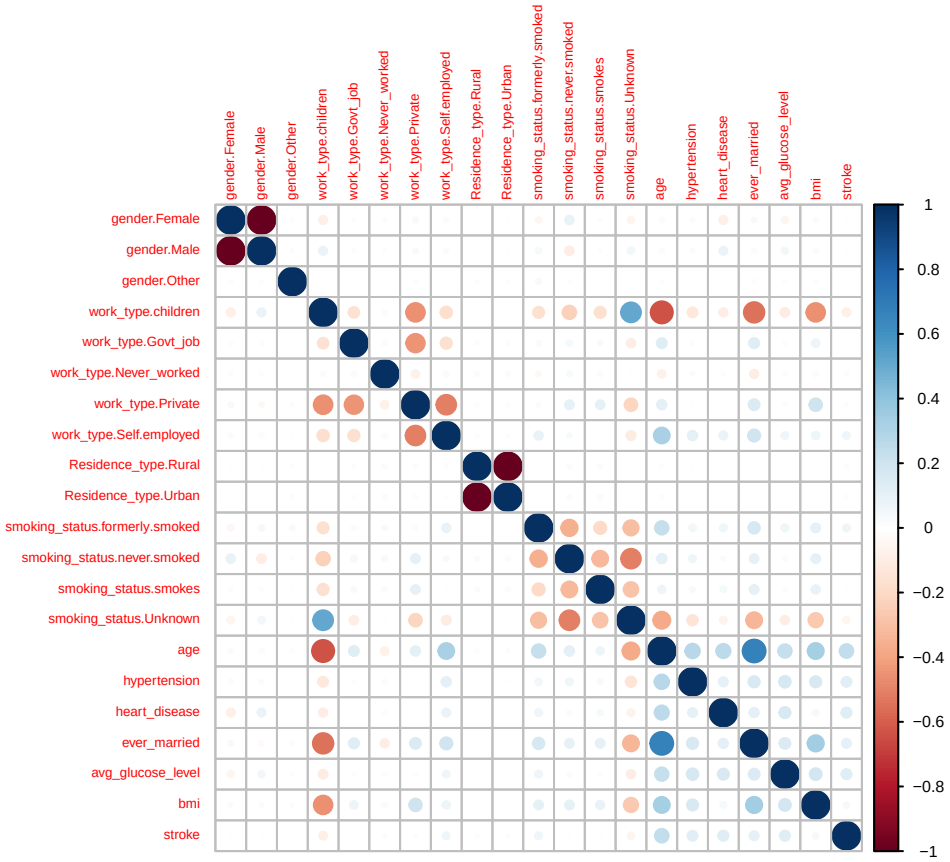
```
strokeDummies$hypertension <- as.numeric(strokeDummies$hypertension)
strokeDummies$heart_disease <- as.numeric(strokeDummies$heart_disease)
strokeDummies$ever_married <- as.numeric(strokeDummies$ever_married)
```

```
par(mar=c(5,5,5,5)+0.1)

# making stroke as numeric to plot correlation
```

```
strokeDummies$stroke <- as.numeric(strokeDummies$stroke)

corrplot(cor(strokeDummies), cl.cex= .5,tl.cex=.4)
```

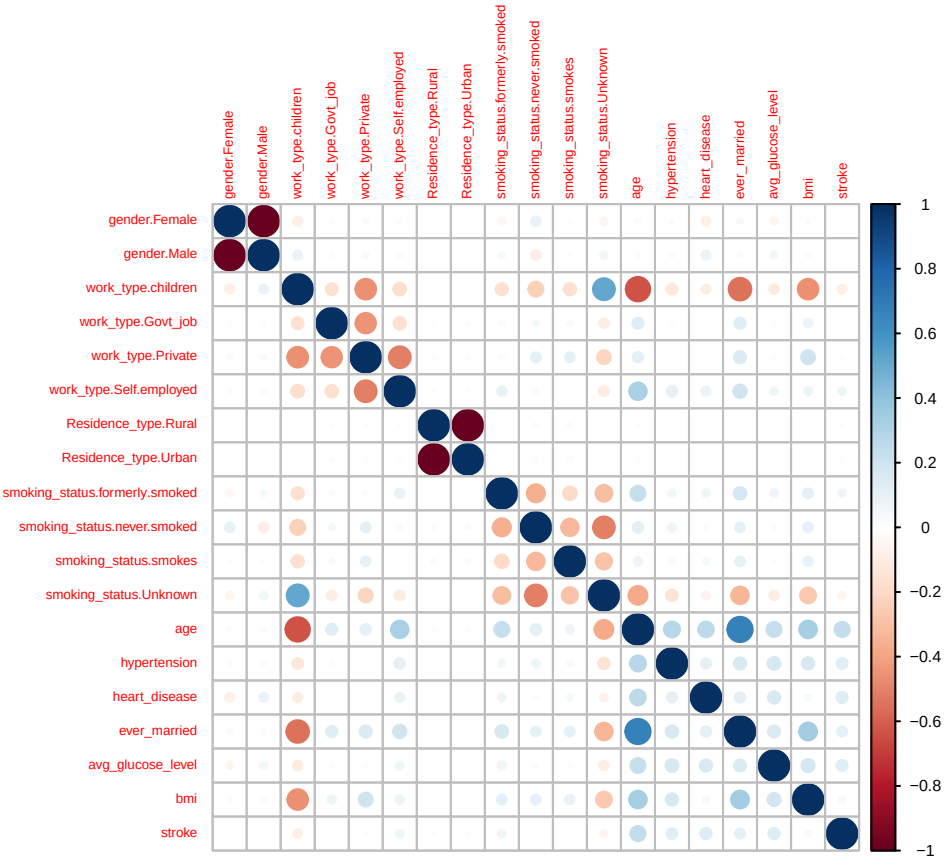


removing nzv predictors and checking correlation again

```
strokeVec <- subset(strokeDummies,select = c(stroke))
strokeImpDum <- preProcess(strokeDummies,"nzv")
strokeDummiesPp <- predict(strokeImpDum, strokeDummies)
```

We lost 3 variables here – gender.Other, work_type.Never_worked, and stroke, which points out how unbalanced the data set is.

```
# add stroke back in
strokeDummiesPp <- cbind(strokeDummiesPp, strokeVec)
corrplot(cor(strokeDummiesPp), cl.cex= .5,tl.cex=.4)
```



Now that we've seen the correlation for each variable, we need to first get rid of any highly correlated features. Then, we need to minimize the dummy variables so we don't have any redundancy IE get rid of one gender (since we only have 2 now) as a 0 for one will imply the existence of the other and so on. First, to get rid of high correlation features using a cut-off of .8.

```
highCorr <- findCorrelation(cor(strokeDummiesPp), cutoff = .8) # find highly correlated predictors
length(highCorr)
```

```
## [1] 2
```

```
filterDummiesPp <- strokeDummiesPp[,-highCorr]
```

2 highly correlated variables were eliminated.

```
colnames(filterDummiesPp)
```

```
##  [1] "gender.Female"                 "work_type.children"
##  [3] "work_type.Govt_job"            "work_type.Private"
##  [5] "work_type.Self.employed"       "Residence_type.Rural"
##  [7] "smoking_status.formerly.smoked" "smoking_status.never.smoked"
##  [9] "smoking_status.smokes"         "smoking_status.Unknown"
## [11] "age"                           "hypertension"
## [13] "heart_disease"                 "ever_married"
## [15] "avg_glucose_level"             "bmi"
## [17] "stroke"
```

In our filtered list of dummies, gender.male was eliminated and so was Residence_type.urban – which took care of part of the other action item. Now, to get rid of one binary feature for the remaining multi-level features to reduce redunancy.

```
filterDummiesPp <- subset(filterDummiesPp, select = -c(smoking_status.Unknown))
```

```
head(filterDummiesPp)
```

```
##   gender.Female work_type.children work_type.Govt_job work_type.Private
## 1             0                  0                  0                 1
## 2             1                  0                  0                 0
## 3             0                  0                  0                 1
## 4             1                  0                  0                 1
## 5             1                  0                  0                 0
## 6             0                  0                  0                 1
##   work_type.Self.employed Residence_type.Rural smoking_status.formerly.smoked
## 1                       0                    0                             1
## 2                       1                    1                             0
## 3                       0                    1                             0
## 4                       0                    0                             0
## 5                       1                    1                             0
## 6                       0                    0                             1
##   smoking_status.never.smoked smoking_status.smokes       age hypertension
## 1                           0                     0 1.0513314            1
## 2                           1                     0 0.7859932            1
## 3                           1                     0 1.6262309            1
## 4                           0                     1 0.2553167            1
## 5                           1                     0 1.5820079            2
## 6                           0                     0 1.6704540            1
##   heart_disease ever_married avg_glucose_level         bmi stroke
## 1             2            2       2.706110617  0.98124492      2
## 2             1            2       2.121350940  0.28606366      2
## 3             2            2      -0.005027809  0.45922236      2
## 4             1            2       1.437217451  0.70113526      2
## 5             1            2       1.501037522 -0.62301952      2
## 6             1            2       1.768021830  0.01359335      2
```

```
# changing stroke back to a factor and ensuring hypertension and heart disease are 0s and 1s
filterDummiesPp$stroke <- as.numeric(as.factor(filterDummiesPp$stroke))
filterDummiesPp$stroke <- ifelse(filterDummiesPp$stroke == 2, "YES","NO")
filterDummiesPp$stroke <- as.factor(filterDummiesPp$stroke)
filterDummiesPp$hypertension <- ifelse(filterDummiesPp$hypertension == 2, 1,0)
filterDummiesPp$heart_disease <- ifelse(filterDummiesPp$heart_disease == 2, 1,0)
filterDummiesPp$ever_married <- ifelse(filterDummiesPp$ever_married == 2, 1,0)
```

```
head(filterDummiesPp)
```

```
##   gender.Female work_type.children work_type.Govt_job work_type.Private
## 1             0                  0                  0                 1
## 2             1                  0                  0                 0
## 3             0                  0                  0                 1
## 4             1                  0                  0                 1
```

```
## 5              1                   0                   0                   0
## 6              0                   0                   0                   1
##   work_type.Self.employed Residence_type.Rural smoking_status.formerly.smoked
## 1                       0                    0                              1
## 2                       1                    1                              0
## 3                       0                    1                              0
## 4                       0                    0                              0
## 5                       1                    1                              0
## 6                       0                    0                              1
##   smoking_status.never.smoked smoking_status.smokes       age hypertension
## 1                           0                     0 1.0513314            0
## 2                           1                     0 0.7859932            0
## 3                           1                     0 1.6262309            0
## 4                           0                     1 0.2553167            0
## 5                           1                     0 1.5820079            1
## 6                           0                     0 1.6704540            0
##   heart_disease ever_married avg_glucose_level         bmi stroke
## 1             1            1       2.706110617  0.98124492    YES
## 2             0            1       2.121350940  0.28606366    YES
## 3             1            1      -0.005027809  0.45922236    YES
## 4             0            1       1.437217451  0.70113526    YES
## 5             0            1       1.501037522 -0.62301952    YES
## 6             0            1       1.768021830  0.01359335    YES
```

Now the data should be ready to model with - it is filtered, centered, scaled, dummified, and reduced of NZV.

Let's split into train and test set

```
trainingRows <- createDataPartition(filterDummiesPp$stroke, p=.8, list=FALSE)
trainStroke <- filterDummiesPp[trainingRows,]

testStroke <- filterDummiesPp[-trainingRows,]

# creating an oversampled training set
set.seed(1103)
trainOs <- upSample(x = trainStroke[1:15],y = trainStroke$stroke,yname = "stroke")

dim(trainStroke)
```

```
## [1] 4089   16
```

```
dim(trainOs)
```

```
## [1] 7778   16
```

```
head(trainStroke)
```

```
##   gender.Female work_type.children work_type.Govt_job work_type.Private
## 2             1                   0                   0                   0
## 3             0                   0                   0                   1
## 4             1                   0                   0                   1
```

```
## 6             0                  0                  0                 1
## 7             0                  0                  0                 1
## 9             1                  0                  0                 1
##   work_type.Self.employed Residence_type.Rural smoking_status.formerly.smoked
## 2                       1                    1                             0
## 3                       0                    1                             0
## 4                       0                    0                             0
## 6                       0                    0                             1
## 7                       0                    1                             0
## 9                       0                    1                             0
##   smoking_status.never.smoked smoking_status.smokes       age hypertension
## 2                           1                     0 0.7859932            0
## 3                           1                     0 1.6262309            0
## 4                           0                     1 0.2553167            0
## 6                           0                     0 1.6704540            0
## 7                           1                     0 1.3608927            1
## 9                           0                     0 0.6975471            0
##   heart_disease ever_married avg_glucose_level         bmi stroke
## 2             0            1       2.121350940  0.28606366    YES
## 3             1            1      -0.005027809  0.45922236    YES
## 4             0            1       1.437217451  0.70113526    YES
## 6             0            1       1.768021830  0.01359335    YES
## 7             1            1      -0.796264185 -0.19012277    YES
## 9             0            1      -0.662440784 -0.25633051    YES
```

There is that heavy imbalance in the training set, so let's balance this dataset. The simplest approach to counteracting the negative effects of class imbalance is to tune the model to maximize the accuracy of the minority class(es). We can do up-sampling or down-sampling, or we can do stratified, or even synthetic minority over-sampling (SMOTE). We will write the train and test sets to csv and start a new notebook for training

```
write.csv(trainStroke,"c:/Users/steph/OneDrive/Documents/USD/ADS503/trainStroke.csv", row.names=F)
write.csv(testStroke,"c:/Users/steph/OneDrive/Documents/USD/ADS503/testStroke.csv", row.names = F)
write.csv(trainOs,"c:/Users/steph/OneDrive/Documents/USD/ADS503/trainStrokeOs.csv", row.names =F)
```